IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Karen R. Kluttz
Sandeep K. Singhal
Thyra Rauch
David A. Schell

GAU: **2179**

Examiner: **Truc T. Chuong**

Application No.: **09/614,852**

Filed: **July 12, 2000**

For: **Method and Apparatus for Determining Computer Interface Display Attributes**

CERTIFICATE OF MAILING/FACSIMILE

I hereby certify that this correspondence, along with any paper indicated as being enclosed, are being deposited with the United States Postal Service as first-class mail, postage pre-paid, on March 28, 2005 in an envelope addressed to: Mail Stop Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

3.28.05
Date

Theodore Naccarella

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attention: Board of Patent Appeals and Interferences

APPELLANTS' BRIEF

This brief is in furtherance of the Notice of Appeal filed in this case on or about January 26, 2005.

1) REQUIRED FEE

The requisite fee of $500.00 set forth in §41.20(b)(2) is submitted herewith and is authorized to be charged to Applicant's Deposit Account No. 09-0461.

03/31/2005 HAHMED1 00000024 090461 09614852
01 FC:1402 500.00 DA

2) REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, a corporation of the State of Delaware, having its principle place of business at New Orchard Road, Armonk, New York 10504, USA. Accordingly, International Business Machines Corporation is the real party in interest.

3) RELATED APPEALS AND INTERFERENCES

The appellant, assignee, and the legal representatives of both are unaware of any other appeal or interference that will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

4) STATUS OF CLAIMS

a. Claims canceled: None

b. Claims withdrawn from consideration but not canceled: None;

c. Claims pending: 1-29

d. Claims allowed: None

e. Claims merely objected to: None

f. Claims rejected: 1-29

g. Appealed Claims: 1-29

Appealed claims 1-29 as currently pending are attached as Appendix A hereto.

5) STATUS OF AMENDMENTS

There are no un-entered amendments to the specification claims or drawings in this case.

6) SUMMARY OF CLAIMED SUBJECT MATTER

The invention claimed in the claims under appeal is best illustrated by Figures 3 and 4 of the present application and described in detail at page 7, line 10 through page 8, line 9 and page 13, line 3 through page 14, line 6. The subject matter of all claims being appealed concerns a method and apparatus for automatically opening files of particular types on a computer using certain attributes, such as window size and window position, as dictated by how the user previously positioned and sized windows when viewing files of the same type. It further includes the concept of, when a user opens a certain file of a first type (the first file), automatically opening a second file that has some particular file name attribute bearing a predetermined relationship to the file name of the first file. For instance, whenever a file having a particular given name with a first file type extension, e.g., johnsmith.doc, is opened, the computer will automatically open a second file having the same file name but a file name extension of a second type, e.g., johnsmith.pdf. The invention is particularly useful for users who repeatedly open one or more files of certain types that they would like to be sized and positioned in the same place every time and/or repeatedly need to open two related files and view them simultaneously, such as might be necessary for repetitive data entry tasks.

In its broadest aspect as recited in independent Claims 1 and 21, the software of the present invention remembers at least one display attribute of a file being used by a user. The attribute, for instance, may be the position and size of the window in which the file is displayed. Then, when the user opens up another file of the same file type, it will open in a window in the same position and of the same size as a previous file of the same type. The invention can be applied to several different file types so that a user can open multiple files that he/she may need to view simultaneously and they will always open up in the position and size windows that the user desires.

7) GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-29 stand rejected under 35 U.S.C. §102(b) as anticipated by U. S. Patent No. 5,613,134 issued to Lucus et al. (hereinafter Lucus).

8) ARGUMENT

A. The Lucus Reference

The Lucus reference pertains to the display of computer files and documents on a computer display. Furthermore, it pertains to the permanent storage of "ephemeral" attributes of a document, including display attributes (such as its coordinates on the display screen) for re-use when the document is displayed a next time. However, contrary to the present invention, Lucus has nothing to do with applying those display

attributes to other files of the same type (which is a core concept of the present invention).

More specifically, Lucus discloses a computer-controlled information management system in which documents in the system have permanent attributes and ephemeral attributes, each attribute having a name and a value. Attributes that are normally permanently stored are called permanent attributes. Attributes that typically are created only when the document is being displayed, such as information regarding the position on the display unit, are called ephemeral attributes. In accordance with the disclosure of Lucus, ephemeral attributes may be converted into permanent attributes and stored with the documents after the user is done referencing or modifying them.

In accordance with another aspect of Lucus that the Examiner deemed significant, Lucus employs a feature referred to as "strands". In accordance with this feature, multiple documents are collected together as a strand and can be treated as a unit. However, again, a strand is a predefined set of documents. There is no discussion whatsoever in Lucus of applying the display attributes of one document to another document.

B. Traversal of Prior Art Rejection

Applicant respectfully traverses the prior art rejections based on Lucus. It should be apparent from the discussion above that Lucus does not teach one of the core concepts of the present invention, namely, applying display attributes that are set in connection with a first document to another document based on that other document being of the same file type as the first document. Originally, in the office action dated

January 29, 2004, in which the Lucus reference was first applied, the Examiner referred to column 9, lines 15-26, column 10, lines 23-54, column 11, lines 17-38, and column 18, lines 45-64 as teaching this claim feature. In the subsequent, Final Office Action dated July 26, 2004, the Examiner additionally referred to column 5, line 56 to column 7, line 3 and Figures 1-3, but dropped any references to column 9, lines 15-26 or column 11, line 17-38. Thus, the Examiner is now relying on column 5, line 56 – column 7, line 3, column 9, line 15-26, column 10, lines 23-54, column 11, lines 17-38, and column 18, lines 45-64.

Let us consider each of these cited portions of Lucus individually. First, column 5, line 56 – column 7, line 3 discusses the storage of permanent and ephemeral attributes in association with a file, as mentioned above. There is no disclosure of applying any of these attributes to any other file but the single file in connection with which it is stored.

Column 9, lines 15-26, column 10, lines 23-54, and column 11, lines 17-38 all discuss the "strand" feature mentioned above, which merely concerns a plurality of files collected as a group and say nothing about applying attributes from one file to another file.

Finally, column 18, lines 45-64 disclose a conventional search tool, such as the search tool used by Windows Explorer. This portion of Lucus merely indicates that files might, for example, be found by file type.

This also is not relevant to the feature of applying attributes assigned to one file to another file.

A review of any of those portions of Lucus reveals no discussion whatsoever of applying attributes of one file to any other file. The cited portions of Lucus discuss applying a given attribute only to the single file with which that attribute was originally associated. There is no suggestion of applying it to any other file, let alone applying it to another file based on file type.

Accordingly, with reference to independent claim 1, Lucus does not teach either of steps (3) and (4), which recite "when another file of the type of said first file is opened by an operator for display, accessing said stored data indicating said value of said at least one attribute" and "displaying said another file of the type of said first file using the same value of said at least one attribute as said first file", respectively.

Dependent claim 12 further distinguishes over Lucus by adding the feature of opening a second file of a certain file type whenever a first file of another file type is opened, the second file having the same defined relationship with the first file as the two files previously displayed simultaneously.

Lucus clearly does not disclose what is claimed in claim 12. While Lucus does appear to disclose opening a group of files with one command, it opens the second and subsequent files simply because they are in a predefined workgroup and not based on file types.

Lucus, therefore, also does not teach the limitations of steps (5) and (6) of dependent claim 12. Particularly, there is nothing in the workgroup concept of Lucus that teaches "storing data associated with said type of said first file indicating at least a type of said second file relative to said first file, as recited in step (5)". Rather, in Lucus, there is simply a list of specific files in a workgroup.

Furthermore, Lucus does not teach "when another file of the type of said first file is opened for display, automatically opening another file of the same type as said second file having the same relationship to said another file of said first type as said second file had to said first file", as recited in step (6). As previously mentioned, Lucus discloses no concept of applying attributes from one file to another file, let alone doing so based on file type.

Independent claim 21 recites subject matter similar to claim 1 and, thus, distinguishes over Lucus for essentially the same reasons as claim 1. Particularly, claim 21 recites "using the same value to display said another file", that value being previously defined as "a value of at least one display attribute of a first displayable file on a computer".

Dependent claim 25 recites subject matter similar to dependent claim 12 discussed above and, thus, even further distinguishes over the prior art for the same reasons given above in connection with claim 12.

All other claims depend from one of claims 1 and 21. Accordingly, they distinguish over the prior art for at least the same reasons.

C. Reply to Examiner's Response to Applicant's Arguments

In the Final Office Action, the Examiner replied to Applicant's arguments set forth above. The Examiner's reply was threefold, addressing Applicant's argument that Lucus does not (1) teach the relationships between documents based on document types; (2) apply attributes of one file to any other file; and (3) display another file using

the same value of the first file. Applicant will herein reply to the Examiner's three points in order.

1. Lucus does not teach the relationships between documents based on document types

In reply to Applicant's argument that Lucus does not teach a relationship between documents based on document type, the Examiner asserted:

> Lucus clearly teaches a Unique Identifier, or UID, is a string of alphanumerics that uniquely identifies a document. A UID is necessary and sufficient to refer to a specific document (col. 4 lines 7-23; the attributes define the display characteristics of an associated document, such as position and size (col. 6 lines 61-66), and by using the UID to define the appearance and location of the documents (col. 10 lines 23-33); therefore, the documents can be retrieved and displayed with the same setup (or layout) of the given set of parameters is matched its type or ID.

The Examiner appears to be arguing that Lucus discloses relating appearance attributes with a document. Applicant does not dispute this. However, this is not what Applicant claims. In Lucus, those attributes are used with the single document with which they are associated. Lucus does not teach taking those attributes that are assigned to one document and applying them to another document.

2. Lucus fails to teach applying attributes of one file to any other file

In addressing this point, the Examiner asserted "Lucus clearly discloses matching types of documents if they are in same characteristics based on UIDs, input parameters, and attributes (e.g., col. 2 lines 18-32)".

However, column 2, lines 18-32 of Lucus state:

> Each time a user requests a document, the client sends a search request to one or more repositories. The repositories respond with one or more messages containing the unique identifier(s) of documents that match the description in the search request. The client then requests permanent attributes of the documents corresponding to the received UIDs, and the repositories respond by sending requested permanent attributes of the requested document to the client. The client then determines whether any of the received permanent attributes for that document are actually ephemeral attributes defining the previous visual display of the document, stored as permanent attributes in the repository. The client converts such permanent attributes into ephemeral attributes and uses their values to create a display of the document on a display device.

The Examiner seems to be misinterpreting what is disclosed in this paragraph of Lucus. As a preliminary point, Applicant is unsure what the Examiner means by the term "matching", but will proceed on the assumption that he is referring to the process of retrieving documents based on their meeting (or "matching") certain search criteria.

The fact that the Examiner has responded to Applicant's argument that "Lucus fails to teach applying attributes of one file to any other file" by asserting that "Lucus clearly discloses matching types of documents if they are in same characteristics based on UIDs, input parameters, and attributes" suggests a fundamental misunderstanding as to what is being claimed because the claimed feature concerning applying attributes of one file to another file does not have anything to do with the alleged teaching of Lucus of matching files by their attributes (or UIDs, for that matter).

As best as Applicant can discern, the Examiner is asserting that Lucus teaches matching documents by file type. Certainly, this is old technology. In fact, any file search software, such as Windows Explorer can "match" files by type. All one would have to do, for instance, is run a Windows Explorer search for files containing a particular file extension.

On the other hand, this paragraph does not teach matching files by attributes or UIDs, as the Examiner seems to be asserting. It merely notes that the documents that matched the search criteria are listed by their UIDs, not that the UIDs are the search criterion.

However, the most troubling thing about the rejection is that, even if Lucus did teach exactly what the Examiner is asserting, it would be entirely irrelevant to the claim element at issue. Specifically, whether or not Lucus teaches retrieving files by their UIDs or display attributes is irrelevant to the claim. The present invention does not have anything to do with retrieving (or matching) files by their display attributes. The present invention concerns displaying a file using the attributes assigned to a different file of the same type. This is nowhere found in Lucus.

The above-quoted paragraph of Lucus discusses two essentially unrelated aspects of Lucus that occur sequentially. However, using hindsight reconstruction, the Examiner seems to be combining them simultaneously in a way that is totally at odds with what is actually disclosed in the paragraph. Specifically, in this paragraph, Lucus describes retrieving a plurality of documents that have something in common with each other (as defined by the query). For instance, they may all be .jpg files. The system returns as a result a list of the UIDS of all of the files that meet the criteria of the search. Then, the client asks for and receives the permanent attributes of the files/UIDs in the list. The client then displays each file using the retrieved attributes for that file. The client does not retrieve the files by these attributes. Even if it did, it would be irrelevant because it would have nothing to do with applying the attributes of one file to any other

file. The attributes are applied only to the single file to which those attributes are assigned.

According to Lucus, each file in the list is displayed using its own attributes. This is the opposite of taking the attributes of one file and using them on another file.

3. Lucus fails to teach displaying another file using the same value of the first file

In addressing this point in the final Office Action, the Examiner asserted "Lucus' invention clearly teaches that every document can be defined by size variables as input parameters which use to determine the size and location the document on the display device (col. 7 lines 26-35); therefore, the other (or second) document will use the same value of the first one when displaying on the screen".

Column 7, lines 26-35 of Lucus are reproduced below for reference:

> For example, every document has a position in world space defined along the x, y, and z axis, and every document has a width and a height. When an image of the document is drawn on the display device, the perspective function takes those world space coordinates and size variables as input parameters, and determines the actual size and location on the display device, in "screen space coordinates", where the document is actually going to be drawn. The perspective function is instantiated by the workspace viewer process.

This paragraph discloses nothing more than that each document is displayed in some defined position on the screen using the display attributes assigned to that document. This paragraph does not discuss how the document positions are defined. While this paragraph does discuss how the size of the document is determined (this appears to be the portion upon which the Examiner is relying), it merely discloses that the "ephemeral" display attributes of each document are used to display that document.

Thus, Lucus does not disclose applying the "ephemeral" display attributes of one document to another document at all, let alone doing so based on the document type. Rather, it teaches applying the "ephemeral" attributes of one document to that exact document and no other.

C. **Distinguishing Claim Language**

With reference to independent claim 1, it should be clear from the discussion above that Lucus does not teach either of steps (3) and (4), which recite "(3) when another file of the type of said first file is opened by an operator for display, accessing said stored data indicating said value of said at least one attribute" and "(4) displaying said another file of the type of said first file using the same value of said at least one attribute as said first file".

Dependent claim 12 further distinguishes over Lucus by adding the feature of opening a second file of a certain file type whenever a first file of another file type is opened, the second file having the same defined relationship with the first file as the two files previously displayed simultaneously.

While Lucus discloses opening a group of files with one command, it opens the second and subsequent files simply because they are in a predefined workgroup and not for any reason relevant to file types.

Lucus, therefore, does not teach the limitations of steps (5) and (6) of claim 12. Particularly, there is nothing in the workgroup concept of Lucus that teaches "(5) storing data associated with said type of said first file indicating at least a type of said second

file relative to said first file". Rather, in Lucus, there is simply a list of specific files in a workgroup.

Furthermore, Lucus does not teach "(6) when another file of the type of said first file is opened for display, automatically opening another file of the same type as said second file having the same relationship to said another file of said first type as said second file had to said first file".

Independent claim 21 recites subject matter similar to claim 1 and, thus, distinguishes over Lucus for essentially the same reasons as claim 1. Particularly, claim 21 recites "using the same value to display said another file", that value being "a value of at least one display attribute of a first displayable file on a computer".

Dependent claim 25 recites subject matter similar to dependent claim 12 discussed above and, thus, even further distinguishes over the prior art for the same reasons given above in connection with dependent claim 12.

All other claims depend from one of claims 1 and 21. Accordingly, they distinguish over the prior art for at least the same reasons.

Respectfully submitted,

Dated: 3.28.05

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

APPENDIX A: CLAIMS INVOLVED IN THIS APPEAL

1. A method of providing an interface with displayable computer files on a computer display, said method comprising the steps of:

(1) displaying a first displayable file on said computer display in a manner customized by an operator of said computer;

(2) storing data indicating a value of at least one attribute of the manner in which said first file was displayed associated with data indicating a type of said first file;

(3) when another file of the type of said first file is opened by an operator for display, accessing said stored data indicating said value of said at least one attribute; and

(4) displaying said another file of the type of said first file using said stored value of said at least one attribute.

2. The method of claim 1 wherein step (2) comprises storing said value when said first file is closed by said operator.

3. The method of claim 2 wherein said value that is stored in step (2) is the value of said attribute when said first file was closed.

4. The method of claim 1 wherein the value that is stored in step (2) is the value of said attribute at a time selected by said operator.

5. The method of claim 1 wherein said attribute is a size of a window within which said first file was displayed.

6. The method of claim 5 wherein said attribute further comprises a position of said window within said computer display.

7. The method of claim 1 wherein said attribute further comprises a position of said window within said computer display.

8. The method of claim 1 wherein said file type comprises a file name extension of said first file.

9. The method of claim 8 wherein said extension comprises multiple extensions.

10. The method of claim 8 wherein step (3) comprises reading said table to determine said attribute.

11. The method of claim 1 wherein said storing step comprises storing said data as an attribute table comprising a plurality of data entries relating file types to attribute values.

12. The method of claim 1 further comprising the steps of:

(4) displaying a second displayable file simultaneously with said first file on said computer display in a manner selected by said operator;

(5) storing data associated with said type of said first file indicating at least a type of said second file relative to said first file; and

(6) when another file of the type of said first file is opened for display, automatically opening another file of the same type as said second file having the same relationship to said another file of said first type as said second file had to said first file.

13. The method of claim 12 wherein step (5) comprises storing said type of said second file when said first file is closed by said operator.

14. The method of claim 13 further comprising the steps of:

(7) storing data indicating a value of at least one attribute of said second file, said data associated with said type of said second file and with said type of said first file; and

(8) when said another file of the same type as said first file is opened by an operator, assigning said another file of the same type as said second file the same value for said attribute of said second file.

15. The method of claim 14 wherein step (7) comprises storing said value when said first file is closed by said operator.

16. The method of claim 15 wherein said value that is stored in step (7) is the value of said attribute when said first file was closed.

17. The method of claim 12 wherein the value that is stored in step (7) is the value of said attribute at a time selected by said operator.

18. The method of claim 14 wherein said attribute of said second file comprises a size of a window within which said second file was displayed.

19. The method of claim 18 wherein said attribute of said second file further comprises a position of said window.

20. The method of claim 14 wherein said attribute of said second file further comprises a position of said window.

21. A computer program for providing an interface with displayable computer files on a computer display, said program comprising:

(1) means for determining a value of at least one display attribute of a first displayable file on a computer;

(2) means for determining a type of said first file;

(3) means for automatically storing said value associated with data indicating said type of said first file when said first file is closed;

(4) means for accessing said stored value when another file of said same type as said first file is opened; and

(5) using the same value to display said another file.

22. The computer program of claim 21 wherein said means for storing comprises means for storing said value of said attribute when said first file was closed.

23. The computer program of claim 22 wherein said file type comprises a file name extension of said first file.

24. The computer program of claim 21 wherein said means for storing comprises means for storing said data as an attribute table comprising a plurality of data entries relating file types to attribute values.

25. The computer program of claim 21 further comprising:

means for determining a value of at least one display attribute of a second displayable file on a computer that is displayed simultaneously with said first file;

means for determining a type of said second file;

means for storing data associated with said type of said first file indicating at least said type of said second file relative to said first file and said at least one attribute of said second file; and

means for automatically opening, when another file of said type of said first file is opened for display, another file of the same type as said second file having the same relationship to said another file of said first type as said second file had to said first file.

26. The method of claim 25 wherein said relationship comprises said first and second files having file names with identical first portions.

27. The method of claim 26 wherein said file names each comprise a first part and an extension part and wherein said file types are dictated by said extension part and said first part comprises said first portion.

28. The computer program of claim 25 wherein said relationship comprises said first and second files having file names with identical first portions.

29. The computer program of claim 28 wherein said file memos each comprise a first part and an extension part and wherein said file types are dictated by said extension part and said first part comprises said first portion.